PARTIEL D'OPTIMISATION COMBINATOIRE 1

Institut Galilée - Université Paris 13 **Partiel du jeudi 25 novembre 2004**
ILOG2 - MMI1 - MI1
Durée : 2 heures, notes de cours et TD non autorisées

1. Exercice 1

Soit le programme linéaire $P(\alpha) = \begin{cases} \max & x_1 + \alpha x_2 \\ s.c. & 5x_1 + x_2 \leq 16 \\ & x_1 + x_2 \leq 7 \\ & x_1 \leq 3 \\ & x_2 \leq 6 \\ & x_1, x_2 \geq 0 \end{cases}$ où $\alpha \in \mathbb{R}$

(1) Résoudre graphiquement le problème pour $\alpha = 1/2$, puis $\alpha = 2$.
(2) Appliquer l'algorithme du simplexe pour $\alpha = 1$ et en déduire l'ensemble des solutions optimales de $P(1)$.

2. Exercice 2

Soit le programme linéaire $(P) = \begin{cases} \max & 2x_1 - x_2 + 8x_3 \\ s.c. & x_4 = 1 - 2x_3 \\ & x_5 = 3 - 2x_1 + 4x_2 - 6x_3 \\ & x_6 = 2 + x_1 - 3x_2 - 4x_3 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{cases}$

(1) Donner une solution triviale du problème.
(2) Faire une itération de l'algorithme du simplexe ; que remarquez-vous ?

3. Exercice 3

Étant donnés $\lambda \in \mathbb{R}$ et $f = (f^1, \ldots, f^n) \in \mathbb{R}^n$ tel que $\exists j \in \{1, \ldots, n\} / f^j > 0$, on s'intéresse au probléme suivant :

$$P(\lambda, f) = \begin{cases} \max & \sum_{j=1}^{n} f^j x_j + \lambda y \\ s.c. & y \sum_{j=1}^{n} x_j = 0 \\ & x_j \in [0,1] \ \forall j = 1, \ldots, n \\ & y \in \{0,1\} \end{cases}$$

Donner les deux valeurs possibles (avec les solutions optimales et réalisables associées) du probléme $P(\lambda, f)$ selon les valeurs relatives du réel λ et du vecteur f.

4. Exercice 4

La transformation d'un système de contraintes d'égalité en un système de contraintes d'inégalité peut se faire simplement en remplaçant chaque contrainte $A_i x = a_i$ par la paire de contraintes $(A_i x \leq a_i,\ A_i x \geq a_i)$, ce qui implique un doublement du nombre de contraintes.

Démontrer qu'il suffit d'ajouter l'unique contrainte

$$(\sum_{i=1}^{m} A_i x \geq \sum_{i=1}^{m} a_i) \ (1) \text{ au système } (A_i x \leq a_i,\ \forall i = 1, \ldots, m) \ (2)$$

pour obtenir l'équivalence avec le système de contraintes initial

$$(A_i x = a_i,\ \forall i = 1, \ldots, m) \ (0)$$

Institut Galilée
Année 2004-2005

Théorie des graphes - Algo3
L3 Informatique - ILOG2

Devoir

A rendre pour le 28/02/2005

1 Sujet

On considère le problème de la recherche des composantes fortement connexes d'un graphe orienté.
Pour résoudre ce problème, nous allons utiliser l'algorithme de Kosaraju-Sharir vu en cours et en TD.

2 Travail à réaliser

Ce devoir est à faire en binôme.

Programmer l'algorithme de Kosaraju-Sharir (ainsi que DFS). Faire une analyse de complexité de votre implémentation. En sortie de votre programme, vous devrez fournir le graphe réduit sous la forme d'un fichier d'instance.

2.1 Bonus

Pour ceux qui désirent aller plus loin, vous pouvez créer une petite interface graphique permettant de sélectionner le fichier d'instance, d'afficher le déroulement de l'algorithme et enfin d'afficher le graphe réduit obtenu.

3 Instances

Vous devrez générer les instances vous permettant de tester votre programme. Pour cela, vous implémenterez un générateur de graphes orientés qui en sortie devra fournir l'instance dans un fichier, ce fichier devant comporter, au minimum, le nombre de sommets, le nombre d'arcs et les arcs (sous forme de matrice d'adjacence ou de liste par exemple). Vous devrez également fournir des explications détaillées du format des instances dans un fichier séparé.

4 A rendre

Le devoir doit contenir un rapport, dans lequel vous devrez décrire le problème, les algorithmes utilisés, les analyses de complexité et les résultats que vous aurez obtenus sur les différentes instances. Votre devoir (et pas le rapport) doit également contenir vos programmes commentés avec makefile et readme si nécessaire, les instances et le fichier explicatif du format des instances. Le rapport est à rendre sous forme de fichier (.txt, .doc, .tex ou .html) et sous forme papier (à déposer dans le casier ILOG2 ou L3 Informatique). Les fichiers (rapport et programmes) sont à envoyer par mel à letocart@lipn.univ-paris13.fr pour les ILOG2 et à toulouse@lipn.univ-paris13.fr pour les L3 Informatique. Le tout doit être mis dans une archive tar.gz ou tgz.

MI1-ILOG2, année 2004-2005
Probabilités et Statistiques
Epreuve Partielle

Exercice 1 On joue n fois avec une pièce de monnaie. La probabilité d'avoir une pile est p, les jets sont supposés indépendants. Soit X_i la variable aléatoire qui vaut 1 si le i-ième jet donne une pile, et 0 sinon. Soient $S = \sum_{i=1}^{n} X_i$ le nombre des piles obtenues, $\overline{X} = \frac{S}{n}$ la proportion des piles.

1. On fixe le nombre de jets n=100 et $p = 0{,}40$.
   (a) Calculer $E(\overline{X})$ et $Var(\overline{X})$. Quelle loi continue peut-on utiliser pour approcher la loi de $\overline{X}$?
   (b) Calculer alors la probabilité que $30 \le S \le 70$.
2. On décide de choisir le nombre de jets n aléatoirement suivant la loi de Poisson de paramètre λ.
   (a) Calculer $P(S=m \mid n=k)$ et $P(S=m, n=k)$ pour $m \le k$.
   (b) Déterminer la loi de S. En déduire $P(n = k|S = m)$.
   (c) Calculer $E(S)$.

Exercice 2 Soit $(X_1,...,X_n)$ un échantillon d'une loi admettant la densité

$$f(x) = \begin{cases} cx^{-1-\frac{1}{\theta}}, \text{ si } x > 1 \\ 0, \quad \text{sinon} \end{cases}, \text{ où le paramètre } \theta > 0.$$

1. Déterminer la constante c en fonction de θ.
2. Calculer $E[\ln(X_1)]$ et $Var[\ln(X_1)]$.
3. Calculer l'estimateur du maximum de vraisemblance $\hat{\theta}$ de θ.
4. Calculer $E(\hat{\theta})$ et $Var(\hat{\theta})$.
5. Calculer l'information de Fisher portée par l'échantillon sur θ.
6. L'estimateur $\hat{\theta}$ est-il sans biais ? Si oui, étudier son efficacité.

Exercice 3 On dispose d'un échantillon $(x_1, x_2,..., x_n)$ de n=18 observations d'une v. a. de loi normale $N(\mu,\sigma^2)$, μ inconnu. Les données suivantes sont extraites de l'échantillon :

$\sum_{i=1}^{n} X_i = 97{,}9$; $\sum_{i=1}^{n} X_i^2 = 582{,}1$.

1. Supposons σ=2. Déterminer un intervalle de confiance pour μ de niveau 95%.
2. Etablir un test de l'hypothèse $H_0: \mu = 6$ contre $H_1: \mu = 5$ au risque de 5%. Quelle est votre décision ? Calculer alors le risque de deuxième espèce β du test.
3. Supposons désormais σ inconnu. Donner une estimation de σ^2. Déterminer un intervalle de confiance pour μ de niveau 95%.

MI1-ILOG2, année 2004-2005
Probabilités et Statistiques
Epreuve Partielle

Exercice 1
On dispose de 3 urnes. La première urne contient 3 boules numérotées 1, 2 et 3 ; la deuxième contient 2 boules numérotées 2 et 3 ; la troisième contient 3 boules numérotées 2, 3 et 4. On choisit au hasard une urne et on tire une boule dans l'urne. Soit U_i l'évènement « la i^{ème} urne est choisie », pour i=1,2,3. On note Y le numéro de la boule tirée.

1. Calculer $P(U_i)$, $P(Y=k|U_i)$, pour $i=1,2,3$, et $1 \le k \le 4$.
2. En déduire la loi de Y et calculer $E(Y)$.
3. Calculer les probabilités conditionnelles $P(U_i|Y=k)$. · $i=1,2$ $k=2,3$
4. Les évènements $Y=1$ et U_3 sont-ils indépendants ? Justifier votre réponse.

Exercice 2
Dans un jeux de dé, on s'intéresse à la sortie du numéro 6 .

1. On jette plusieurs fois le dé. On suppose que les jets sont indépendants. A est l'évènement « exactement un 6 au cours des k-1 premiers jets ($k>1$) », B l'évènement « le k-ième jet donne un 6 ». Quelle est la probabilité de A et de B.
2. Soit X la variable aléatoire qui est le nombre minimum de jets nécessaires pour obtenir deux 6. Exprimer l'événement $X=k$ en fonction de A et de B.
3. En déduire la loi de X.
4. Calculer $E(X)$ et $Var(X)$.(Indication : si $|\alpha|<1$, $\sum_{n=0}^{\infty}\alpha^n = \frac{1}{1-\alpha}$,

$$\sum_{n=1}^{\infty} n\alpha^{n-1} = \frac{1}{(1-\alpha)^2},\ \sum_{n=2}^{\infty} n(n-1)\alpha^{n-2} = \frac{2}{(1-\alpha)^3},\ \sum_{n=3}^{\infty} n(n-1)(n-2)\alpha^{n-3} = \frac{6}{(1-\alpha)^4})$$

Exercice 3
On tire successivement et sans remise 2 boules dans une urne qui contient n boules numérotées de 1 à n. Y_1 le premier numéro tiré et Y_2 le second numéro.

(1) Déterminer la loi jointe de (Y_1,Y_2) ainsi que les lois marginales de Y_1 et de Y_2.
(2) Soit $Z=\max(Y_1,Y_2)$, calculer $P(Z \le k)$, pour $1 \le k \le n$.
(3) En déduire la loi de Z. Calculer $E(Z)$.
(4) Soit $U=\min(Y_1,Y_2)$. Déterminer la loi de U et calculer $E(U)$.

(Indication : $\sum_{k=1}^{n} k^2 = \frac{1}{6}n(n+1)(2n+1)$)

UNIVERSITE PARIS 13	Année Universitaire
Institut Galilée	2004 / 2005
ILOG2	Bases de Données
	F. Boufarès
Durée : 3 heures	14 Juin 2005
Tous les documents sont autorisés	

EXERCICE 1 : La gestion commerciale GESCOM

Le système d'information de l'entreprise BB BOBO est articulé autour de la base de données GESCOM décrite par le schéma relationnel ci-dessous.

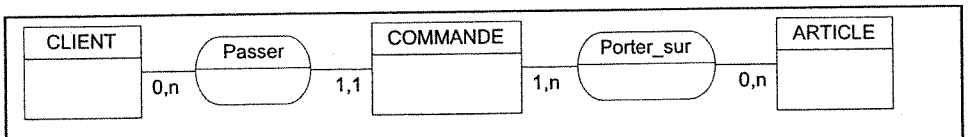


Schéma conceptuel de données

Pour la gestion de ses commandes, les articles sont identifiés de manière unique par la référence (RefArt). Les clients sont identifiés par le code (CodCli). Les commandes sont retrouvées grâce au numéro (NumCom). Une ligne de comande est identifiée par la combinaison NumCom et RefArt.

CLIENTS (**CodCli**, CivCli, NomCli, PrenCli, CatCli, AdnCli, AdrCli, CPCli, VilCli, PaysCli)
COMMANDES (**NumCom**, ***CodCli****, DatCom)
ARTICLES (**RefArt**, NomArt, PVArt, QsArt, PAArt)
DETAILCOM (***NumCom*, *RefArt****, QtCom, PUArt, Remise)

Schéma relationnel de la BD GESCOM

Questions : 2+4+4=10 points → SQL & PL/SQL

1.1. Formulez en SQL la requête qui permet de déterminer le chiffre d'affaires par pays.
1.2. Développez le mécanisme (un trigger, la ou les tables nécessaires ...) qui permet de garder un historique des prix de ventes (PVArt) et des prix d'achat (PAArt). Lors des mises à jour de la table ARTICLES on veut garder une trace des prix...
1.3. Développez une procédure PL/SQL qui permet de vérifier la cohérence de la base de données : "Vérifier si toutes les commandes portent au moins sur un article et afficher celles qui sont donc vides !"

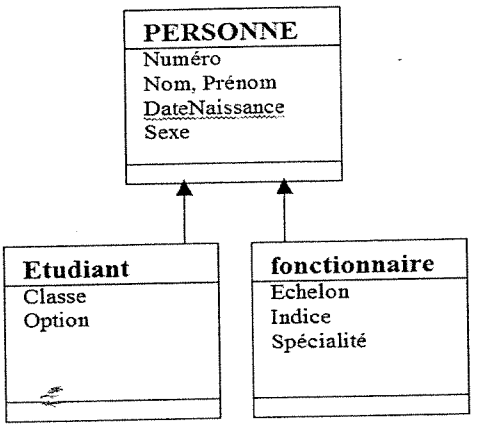
Le dictionnaire de données de la BD GESCOM est le suivant :

Attribut/Colonne	Signification / Description	Type de données / Contrainte
AdnCli	Adresse du client (Numéro)	Caractères/Texte, 10
AdrCli	Adresse du client (Rue/Avenue/Boulevard …)	Caractères/Texte, 50
CatCli	Catégorie du client	Entier, compris entre 1 et 7
CivCli	Civilité du client	Mademoiselle, Madame, Monsieur
CodCli	Code du client	Caractères/Texte, 5
CPCli	Code postal du client	Caractères/Texte, 10
DatCom	Date de la commande	jj-mmm-aaaa
NomArt	Nom (désignation) de l'article	Caractères/Texte, 50
NomCli	Nom du Client	Caractères/Texte, 20
NumCom	Numéro de la commande	Caractères/Textes, 10
PAArt	Prix d'achat de l'article	Réel, 10 dont 2 décimales
PaysCli	Pays où habite le client	Caractères/Texte, 30, en MAJUSCLUE
PrenCli	Prénom du client	Caractères/Texte, 20
PVArt	Prix de vente de l'article, Prix catalogue	Réel, 10 dont 2 décimales
PUArt	Prix de vente de l'article à la date de commande	Réel, 10 dont 2 décimales
QsArt	Quantité en stock de l'article	Entier, 3 (<1000)
QtCom	Quantité commandée	Entier, 3 (<1000)
RefArt	Référence de l'article	Caractères/Texte, 10
Remise	Taux de remise appliqué à l'article	Réel, 4 dont 2 décimales
VilCli	Ville où habite le client	Caractères/Texte, 20, en MAJUSCLUE

EXERCICE 2 :

Questions : 2+3=5 points → SQL2 et SQL3

Considérons le schéma suivant présentant des liens d'héritage



2.1. Traduisez les contraintes d'héritage spécifiées ci-dessus en utilisant SQL2 .
2.2. Traduisez les contraintes d'héritage spécifiées ci-dessus en utilisant SQL3 .

EXERCICE 3 :

Questions : 1+2=3 points → L'Objet en SQL3

3.1.Créez un type objet cours .
3.2. créez ensuite une table d'objets cours.

.

EXERCICE 4 :

Question : 2 points → Les entrepôts de données

4.1.Donnez la définition des entrepôts de données et citez leurs spécificités par rapport aux bases de données transactionnelles

Bonne chance

Institut Galilée - Université Paris 13
ILOG2

OC2
Semaine du 31 mai 2005

T.P. Examen
Durée : 2 heures
Notes de cours, de TD et de TP autorisées

Planification d'examens

Dans une école d'ingénieurs de la région parisienne, chaque semestre, chaque étudiant de deuxième année choisit un ensemble de huit modules parmi onze proposés, en fonction de l'option qu'il désire suivre en troisième année. Ces options sont au nombre de deux : "Aide à la décision et optimisation" et "Communication homme-machine et documents électroniques". Pour le semestre courant, certains modules sont obligatoires pour les étudiants désirant s'orienter vers ces deux options. Il s'agit des modules de Probabilités (P), Algorithme et complexité III (AC), Ingénierie des connaissances (IC) et Systèmes distribués et parallélisme (SDP). Certains autres sont optionnels : Bases de données II (BD), Réseaux (R), Optimisation combinatoire II (OC), Interface graphique (IG), C++ (C++), Paradigme logique (PL) et Génie logiciel III (GL).

	BD	R	C++	GL	IC	IG	AC	PL	OC	P	SDP
BD		x			x		x			x	x
R	x				x		x			x	x
C++				x	x	x	x		x	x	x
GL			x		x	x	x			x	x
IC	x	x	x	x		x	x	x	x	x	x
IG			x	x	x		x		x	x	x
AC	x	x	x	x	x	x		x	x	x	x
PL					x		x			x	x
OC			x		x	x	x			x	x
P	x	x	x	x	x	x	x	x	x		x
SDP	x	x	x	x	x	x	x	x	x	x	

Mme Eudété doit planifier les examens prévus à la fin du semestre. Chaque examen dure deux heures. Deux journées sont réservées pour planifier ces examens suivant les plages horaires suivantes : 8h-10h, 10h15-12h15, 14h-16h et 16h15-18h15. Pour chaque examen, elle connaît l'ensemble des examens incompatibles, qui ne peuvent avoir lieu en même temps car devant être effectués par des étudiants communs. Ces incompatibilités sont résumées dans le tableau, une croix indiquant une incompatibilité.
Aidez Mme Eudété à construire un emploi du temps de telle sorte qu'aucun étudiant n'ait plus d'un examen en même temps.

T.P. Examen
Durée : 2 heures
Notes de cours, de TD et de TP autorisées

Planification d'examens

Dans une école d'ingénieurs de la région parisienne, chaque semestre, chaque étudiant de deuxième année choisit un ensemble de huit modules parmi onze proposés, en fonction de l'option qu'il désire suivre en troisième année. Ces options sont au nombre de deux : "Aide à la décision et optimisation" et "Communication homme-machine et documents électroniques". Pour le semestre courant, certains modules sont obligatoires pour les étudiants désirant s'orienter vers ces deux options. Il s'agit des modules de Probabilités (P), Algorithme et complexité III (AC), Ingénierie des connaissances (IC) et Systèmes distribués et parallélisme (SDP). Certains autres sont optionnels : Bases de données II (BD), Réseaux (R), Optimisation combinatoire II (OC), Interface graphique (IG), C++ (C++), Paradigme logique (PL) et Génie logiciel III (GL).

	BD	R	C++	GL	IC	IG	AC	PL	OC	P	SDP
BD		x			x		x			x	x
R	x				x		x			x	x
C++				x	x	x	x		x	x	x
GL			x		x	x	x			x	x
IC	x	x	x	x		x	x	x	x	x	x
IG			x	x	x		x		x	x	x
AC	x	x	x	x	x	x		x	x	x	x
PL					x		x			x	x
OC			x		x	x	x			x	x
P	x	x	x	x	x	x	x	x	x		x
SDP	x	x	x	x	x	x	x	x	x	x	

Mme Eudété doit planifier les examens prévus à la fin du semestre. Chaque examen dure deux heures. Deux journées sont réservées pour planifier ces examens suivant les plages horaires suivantes : 8h-10h, 10h15-12h15, 14h-16h et 16h15-18h15. Pour chaque examen, elle connaît l'ensemble des examens incompatibles, qui ne peuvent avoir lieu en même temps car devant être effectués par des étudiants communs. Ces incompatibilités sont résumées dans le tableau, une croix indiquant une incompatibilité.
Aidez Mme Eudété à construire un emploi du temps de telle sorte qu'aucun étudiant n'ait plus d'un examen en même temps.

Institut Galilée
Année 2004-2005

Systèmes Distribués
ILOG2

Devoir de Systèmes Distribués

Lundi 11 04 2005

RMI

Le but du devoir est la mise en oeuvre d'un système basé sur l'architecture RMI pour invoquer les méthodes (fonctions) à distance (l'ancêtre étant le RPC).Vous procéderez ainsi :

1. Parmi tous les algorithmes qui vous ont été présentés en cours TD-TP, vous en choisissez deux ou trois en mode centralisé que vous programmerez en JAVA en tant que méthodes invocables à distance.
2. Vous mettez les machines en anneau sous RMI (au moins 3 machines).
3. Vous implantez la méthode de Chang & Roberts pour l'élection d'un serveur qui se fera connaitre sur l'anneau.
4. Les machines battues sont clientes et peuvent invoquer les méthodes mises en service sur le serveur. Bien-entendu, les utilisateurs sur ces machines clientes ont droit à un affichage graphique soigné.
5. *Challenge* : Les machines clientes peuvent également aller déposer leurs méthodes chez le serveur et faire profiter à tous de leur génie créateur. Ces méthodes deviennent invocables pour tous !.
6. Le logiciel (complet, professionnel quoi ! en .html ou .pdf ou ...) doit m'être envoyé par e-mail à *jc@lipn.univ-paris13.fr*. Je dois y trouver entre autres, une analyse en complexité de la solution Chang & Roberts dans le cas général où il y a plus d'un initiateur, un manuel d'installation de votre logiciel, et un *readme* pour le lancement du programme. Une démo de votre logiciel sera demandée.

EXAMEN DE GESTION – GROUPE 1

Mercredi 15 décembre 2004

Question 1 : En plus d'une activité technique, l'ingénieur a de plus en plus une activité managériale. Après avoir défini l'expression activité managériale, exposez une raison expliquant le sens de cette phrase (2 points).

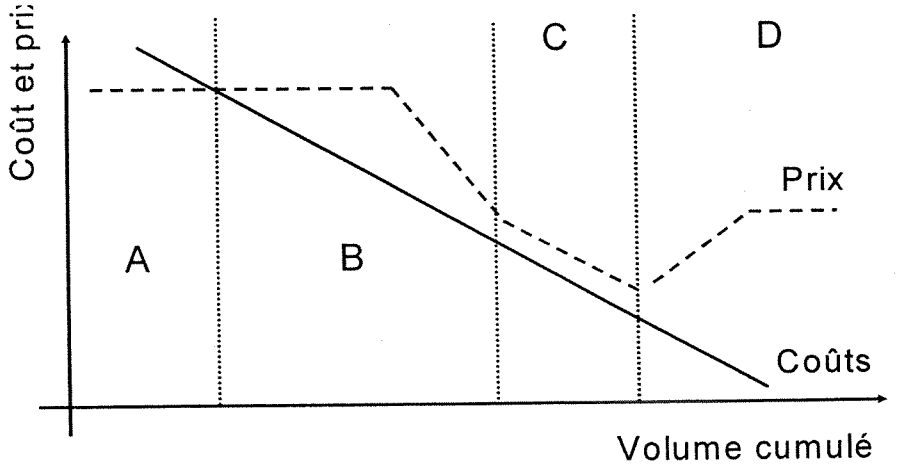
Question 2 : Pourquoi les restructurations dans les grandes entreprises au cours des années 1980 et 1990 se sont-elles accompagnées d'un mouvement de déconcentration ? Avant de répondre à cette question, vous préciserez la signification du terme déconcentration (4 points).

Question 3 : Quelles sont les différentes formes que peut prendre un investissement socialement responsable ? Citez et définissez-en deux (2 points).

Question 4 : Selon la grille d'analyse proposée par M. PORTER, cinq forces commandent la concurrence et déterminent conjointement la rivalité et la rentabilité d'un secteur d'activité. Citez et définissez en quatre (2 points).

Question 5 : Pourquoi est-il équivalent de parler de stratégie de coût et de stratégie de volume ? (1 point)

Question 6 : Définissez et commentez les différentes phases du graphique ci-dessous (4 points) :



Question 7 : En quoi consiste une stratégie de différenciation ? Définissez et donnez deux exemples de stratégie de différenciation (3 points).

Question 8 : Quelle est l'utilité pour une entreprise de disposer d'un modèle de gestion de son portefeuille d'activité ? Citez trois limites des modèles présentés en cours ? (2 points)

EXAMEN DE GESTION – GROUPE 1

Mercredi 9 mars 2005

Question 1 (3 points) : Si la stratégie détermine la structure, la structure conditionne également la stratégie. Exposez brièvement trois raisons justifiant que les dirigeants tiennent comptent de la structure dans l'élaboration d'une stratégie globale.

Question 2 (2 points) : Définissez et schématisez une structure divisionnelle simple.

Question 3 (3 points) : L'évolution de l'économie capitaliste, depuis le marché de pénurie du début du XIX^{ème} siècle jusqu'au marché de concurrence internationale de la fin du XX^{ème} siècle a fortement influencé l'organisation interne de l'entreprise. La fonction commerciale y a pris une place de plus en plus importante. Généralement, on distingue trois phases dans cette évolution. Pouvez-vous nommez et définir ces trois phases ?

Question 4 (2 points) : Quelles sont les principales méthodes de fixation des prix auxquelles recourent les entreprises ?

Question 5 (2 points) : En dépit de l'informatisation et de l'automatisation de la production, le travail garde un rôle essentiel dans la création de biens et de services par les entreprises. Mais le travail a évolué dans son contenu et les objectifs qui sont assignés aux travailleurs ont changé : le « toujours plus » a cédé la place au « toujours mieux ». Quels sont les deux principaux facteurs de l'évolution en cours ?

Question 6 (1 point) : En quoi consiste l'annualisation du temps de travail ?

Question 7 (2 points) : Pour faire face à une période difficile et sans recourir à des licenciements, quels sont les autres procédés à la disposition d'une entreprise pour réduire ses effectifs ?

Question 8 (2 points) : Quelle différence faites-vous entre qualification et compétence ?

Question 9 (3 points) : Après avoir défini le besoin en fonds de roulement, vous expliquerez pourquoi l'activité d'une entreprise engendre un tel besoin. Vous pourrez illustrer votre réponse par un schéma.

Question 10 (2 points) : Quels sont les principaux moyens externes dont dispose une entreprise pour financer sont cycle d'investissement ?

Partiel d'Ingénierie des Connaissances
jeudi 9 juin 2005
9h – 11h
Amphi Euler

Tous documents manuscrits autorisés
Barême indicatif

1 Analyse syntaxique de texte en LN (5 pts)

Soit le texte suivant :
« Unissons-nous contre le projet d'implantation d'un Centre de Rétention Administrative à Deuil-la-Barre. Unissons-nous afin que notre ville reste toujours UNE VILLE DANS SON JARDIN.»
(extrait de la pétition -NON au CRA- du maire de Deuil-la-Barre, J-C Noyer, juin 2005)

1.1 Considérez les noms ou expressions suivants : *implantation, Centre, Rétention, Administrative, Deuil-la-Barre, jardin.* Indiquez à quels mots ils se rattachent pour permettre une analyse sémantique correcte. Par exemple, *Administrative* se rattache à *Rétention*, pas à *implantation ;* par contre, *Deuil-la-Barre* se rattache à *implantation.*

1.2 On propose le lexique suivant : (remarque : *d'=de*)

Noms : *projet, implantation, Centre, Rétention, Deuil-la-Barre, ville, jardin*
Adjectifs *: Administrative*
Verbes : *Unissons-nous, reste*
Conjonction de subordination: *afin que*
Adverbe *: toujours*
Déterminants/articles : *le, un, notre, une, son*
Prépositions : *contre, d', à, dans*

et la grammaire suivante (GN=groupe nominal, GP=groupe prépositionnel)

Phrase -> Verbe GP | Verbe ConjSub Phrase | GN Verbe Adv GN
GN-> GN GP | GN GP GP |[Déterminant] Nom [Adj]
GP -> Préposition GN | Préposition Nom

Montrez comment cette grammaire permet d'analyser les phrases données de manière à retrouver le découpage précédent. Donnez pour chaque phrase l'arbre syntaxique résultant de cette analyse.

1.3 Montrez que la grammaire proposée permet d'analyser la première phrase d'autres manières. Expliquez quels problèmes va rencontrer un analyseur syntaxique pour automatiser le découpage.

1.4 Comment pourrait-on améliorer la grammaire pour empêcher ces mauvaises analyses ?

2 Apprentissage de concepts : construction de l'espace des versions (7 pts)

2.1 Questions de cours (2 pts) :

Qu'est ce qu'une généralisation minimale ?
Qu'est ce qu'une spécialisation maximale ?
Donner 2 règles de généralisation.
Donner 2 règles de spécialisation

2.2 Exercice (5pts)

On décrit des pierres par leur forme (ronde, oblongue, irrégulière), leur poids (en gr), le fait qu'elles

soient coupantes ou pas. Ces pierres sont peut-être des outils préhistoriques. On veut apprendre à reconnaître les outils. Ces objets sont donc soit des exemples positifs d'outils (+), soit des exemples négatifs (-). On va construire l'espace des versions définissant l'ensemble de tous les concepts cohérents avec les exemples positifs et négatifs connus, c'est à dire dont les éléments acceptent les outils et rejettent les autres pierres. Soient les exemples suivants :

e1+ (oblongue $\wedge$ 300 $\wedge$ non-coupante)
e2- (irrégulier $\wedge$ 700 $\wedge$ non-coupante)
e3+ (irrégulier $\wedge$ 400 $\wedge$ coupante)
e4- (ronde $\wedge$ 60 $\wedge$ non-coupante)
e5+ (oblongue $\wedge$ 150 $\wedge$ coupante)

2.2.1 Déroulez l'algorithme pas à pas en donnant à chaque fois G et S les ensembles des hypothèses respectivement les plus générales et les moins générales de H, ensemble des hypothèses. Lors de l'initialisation, G = {vrai} et S = $\varnothing$. Expliquez à chaque modification de G ou S quelles spécialisations maximales ou généralisations minimales vous introduisez.

2.2.2 Donnez un exemple de concept plus général qu'un élément de S et plus spécifique qu'un élément de G, qui sera donc solution.

3 Construction d'ontologie (8 pts)

3.1 Construisez une ontologie en logique de description (CLASSIC, dont le langage est donné en annexe) qui décrive les connaissances ci-dessous. Conservez autant que possible les termes du texte pour désigner les concepts et les rôles. Précisez bien quels sont les concepts (génériques) et les individus (concepts individuels). Choisissez et respectez une convention de nommage : par exemple, les noms de concepts en majuscule en concaténant les mots comme ENTITEABSTRAITE, les noms d'individus avec une majuscule comme JeanDupont et les noms de rôles en minuscule comme aPourDate.

- entité et processus sont les concepts de plus haut niveau (juste sous topConcept)
- une entité administrative est une entité abstraite
- un bâtiment est une entité concrète
- une personne est une entité concrète
- une entité administrative se situe dans un bâtiment
- un collégien est une personne
- un lycéen est une personne
- un étudiant est une personne
- un collégien est élève dans un collège
- un lycéen est élève dans un lycée
- un étudiant est élève dans une université
- un collège, un lycée, une université sont des entités administratives qui délivrent des diplômes
- un lycée délivre le baccalauréat
- un diplôme possède les caractéristiques aPourDate, aPourDiscipline et aPourMention
- toute personne peut avoir un diplôme

3.2 Complétez les connaissances ci-dessus pour que votre ontologie soit cohérente (qu'il ne manque pas de relations et que toutes les relations soient correctes).

3.3 Rajoutez les connaissances nécessaires (concepts et rôles génériques ou individuels) pour représenter le fait que l'individu Jean Dupont a obtenu son baccalauréat S avec mention Bien en juillet 2001.

3.4 Quelles connaissances sur l'individu Jean Dupont un classifieur peut-il déduire ?

3.5 Précisez la cardinalité des rôles seSitue-Dans et aPourDate.

3.6 Comment pourriez-vous exprimer qu'une personne a un diplôme si et seulement si elle a suivi les études correspondantes (en particulier, elle a été élève dans un lycée pour avoir le baccalauréat)?

3.7 Proposez une modélisation plus raffinée qui permette de détailler les diplômes délivrés par le département d'informatique de l'Institut Galilée.

Logique de description

Langage Classic

1.Vocabulaire

Ensemble de symboles non logiques

- Symboles de concept (CONCEPT)
- Symboles d'individus (Individu)
- Symboles de rôles (rôle)

Ensembles de symboles logiques

AND, ALL, FILLS, AT-MOST, AT-LEAST, ONE-OF, SAME-AS, PRIMITIVE, DISJOINT-PRIMITIVE, INVERSE, ATTRIBUT.

Vocabulaire auxiliaire

DefConcept, DefIndividu, DefRôle, DescrConcep, DescrIndividu, DescrRôle, DescrAttribut, DescrRègle, NomConcept, NomIndividu, NomRôle, NomAttribut, CombConcept, Définition, DéfinitionExtensionnelle, DéfinitionPrimitive, Restriction, RestrictionCardinalité, RestrictionValeur, CheminAttributs, EntierPos

2.Syntaxe

DefConcept -> *NomConcept* = *DescrConcept*
DefIndividu -> *NomIndividu* = *DescrConcept*
DefRôle -> *NomRôle* = *DescrRôle*
DescrConcept -> *NomConcept* | *CombConcept* | *Définition* | *Restriction*
CombConcept -> (AND *DescrConcept* +)
Définition -> *DéfinitionExtensionnelle* | *DéfinitionPrimitive*
DéfinitionExtensionnelle -> (ONE-OF *NomIndividu* +)
DéfinitionPrimitive -> (PRIMITIVE *DescrConcept*) |
(DISJOINT-PRIMITIVE *DescrConcept* +)
Restriction -> *RestrictionCardinalité* | *RestrictionValeur*
RestrictionCardinalité -> (AT-LEAST *EntierPos NomRôle*) |
(AT-MOST *EntierPos NomRôle*)
RestrictionValeur -> (ALL *NomRôle DescrConcept*) |
(FILLS *NomRôle NomIndividu* +) |
(SAME-AS *CheminAttributs CheminAttributs*) |
DescrIndividu -> *DescrConcept*
DescrRôle -> *NomRôle* | (INVERSE *NomRôle*) |
DescrAttribut -> (ATTRIBUT *NomRôle*)
DescrRègle -> *NomConcept* => *DescrConcept*
CheminAttributs -> (*NomAttribut* *)
NomAttribut -> 1 nom de rôle
NomConcept -> 1 symbole de concept
NomIndividu -> 1 symbole d'individu
NomRôle -> 1 symbole de rôle
EntierPos -> 1 entier positif

EXAMEN D'OPTIMISATION COMBINATOIRE 1

Institut Galilée - Université Paris 13 **Examen du lundi 24 janvier 2005**
ILOG2 - MMI1 - MI1
Durée : 3 heures, notes de cours et TD autorisées

1. Exercice 1

Soit le programme linéaire $\mathcal{P} = \begin{cases} \max & 2x_1 + x_2 & & \\ s.c. & 4x_1 - x_2 & \leq & 8 \\ & -x_1 + x_2 & \leq & 1 \\ & 3x_1 + 2x_2 & \geq & 6 \\ & x_1, x_2 \geq 0 & & \end{cases}$

(1) Résoudre graphiquement le problème $\mathcal{P}$.
(2) Dérouler la première phase de l'algorithme primal du simplexe en suivant le critère de Dantzig. (choisi de d)

Indication : le dernier tableau du simplexe (correspondant à la base réalisable optimale de $\mathcal{P}$) est le suivant

Base	t	T^1	T^2	T^3	T^4	T^5	T^6
A^5	11	0	0	5/3	11/3	1	-
A^2	4	0	1	1/3	4/3	0	-
A^1	3	1	0	1/3	1/3	0	-
	-10	0	0	-1	-2	0	-

2. Exercice 2

Soit le programme linéaire $\mathcal{Q} = \begin{cases} \max & 2x_1 + x_2 & & \\ s.c. & 4x_1 - x_2 & \leq & 8 \\ & -x_1 + x_2 & \leq & 1 \\ & x_1, x_2 \geq 0 & & \end{cases}$

(1) Que pouvez-vous dire a priori sur les valeurs optimales relatives de $\mathcal{P}$ et de $\mathcal{Q}$?

Le dernier tableau du simplexe (correspondant à la base réalisable optimale de $\mathcal{Q}$) est le suivant

Base	t	T^1	T^2	T^3	T^4
A^2	4	0	1	1/3	4/3
A^1	3	1	0	1/3	1/3
	-10	0	0	-1	-2

(2) Dans quelle mesure peut-on faire varier le paramètre λ sans changer de base réalisable optimale I^* :
   (a) Si l'on remplace la contrainte $4x_1 - x_2 \leq 8$ par la contrainte $4x_1 - x_2 \leq 8 + \lambda$?
   (b) Si l'on remplace la contrainte $-x_1 + x_2 \leq 1$ par la contrainte $-x_1 + x_2 \leq 1 + \lambda$?
   (c) Si la fonction objectif $2x_1 + x_2$ devient la fonction $(2 + \lambda)x_1 + x_2$?
   (d) Si la fonction objectif $2x_1 + x_2$ devient la fonction $x_1 + (1 + \lambda)x_2$?

Pour chacune des quatre questions précédentes, donner la solution de base $x(I^*, \lambda)$ et sa valeur $v_{\mathcal{Q}}(x(I^*, \lambda))$ en fonction de λ.

(3) Écrire de dual $\mathcal{D}_{\mathcal{Q}}$ du prolème $\mathcal{Q}$.
(4) Résoudre $\mathcal{D}_{\mathcal{Q}}$ des quatre manières suivantes :
   (a) résolution graphique du problème dual ;
   (b) détermination de la solution duale à partir de la fonction objectif et du tableau optimal ;
   (c) déduction à partir du vecteur de coûts marginaux de la base optimale.
   (d) déduction à partir des relations d'exclusion.

3. Exercice 3

Soit le programme linéaire $(\mathcal{R}) = \begin{cases} \min & 5x_1 + 3x_2 & \\ s.c. & 4x_1 - x_2 & \leq 8 \\ & -x_1 + x_2 & \leq 1 \\ & 3x_1 + 2x_2 & \geq 6 \\ & x_1, x_2 \geq 0 & \end{cases}$

(1) Initialiser l'algorithme dual du simplexe.
(2) Faire une seule itération de l'algorithme dual du simplexe.

4. Exercice 4

Soit le programme linéaire $(\mathcal{S}) = \begin{cases} \min & -3x_1 + x_2 - 2x_3 & \\ s.c. & x_1 - x_2 + 2x_3 & \leq 10 \\ & -x_1 + x_2 & \geq 3 \\ & x_1, x_2, x_3 \geq 0 & \end{cases}$

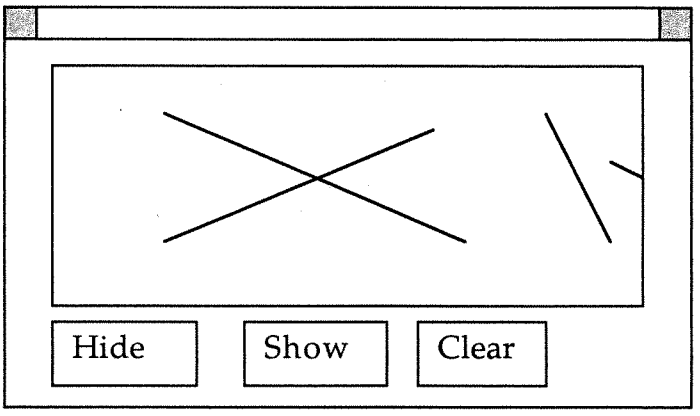
(1) Résoudre $\mathcal{S}$ par l'algorithme primal du simplexe.
(2) Écrire le dual $\mathcal{D}_{\mathcal{S}}$ de $\mathcal{S}$; confirmer le résultat obtenu à la question précédente par résolution graphique de $\mathcal{D}_{\mathcal{S}}$.

C. RECANATI
Interfaces Graphiques
ILOG 2, 2004-2005

Examen du 21 octobre 2004

(14h-17h)

On veut écrire un programme présentant une fenêtre dans laquelle on pourra dessiner des lignes. Cette fenêtre sera disposée au dessus de trois boutons alignés, comme sur la figure ci-dessous (les boutons sont implantés ici par des fenêtres dans lesquelles ont dessine une chaîne de texte) qui permettront de lancer des commandes.



Lors d'un changement de taille de la fenêtre principale, la fenêtre de dessin de lignes devra s'agrandir afin d'occuper le maximum d'espace en largeur et en hauteur - en laissant, bien entendu, l'espace nécessaire en bas aux trois fenêtres boutons. Les fenêtres boutons seront donc attachées comme il convient au bas de la fenêtre principale pour qu'il n'y ait pas besoin de les repositionner.

On pourra à tout moment dessiner des lignes dans la fenêtre graphique de la façon suivante : on clique en un point, et, en maintenant le bouton de souris enfoncé, on tire la souris jusqu'à un autre point où l'on relâche le bouton. Pendant tout le temps du tirer de la souris, la ligne joignant le point initial et le point courant est affichée interactivement (la précédente étant effacée à chaque fois). La dernière ligne tracée (i.e. celle tracée au moment du relâcher) sera également dessinée dans un Pixmap qui sauvegardera ainsi le dessin des lignes. Ce Pixmap sera utilisé pour réafficher les droites avec la commande du bouton Show, et pour traiter les événements de type Expose pouvant survenir sur la fenêtre de dessin.

Programmer le sujet dans l'ordre suivant :

1) Créez et affichez les différentes fenêtres. Veillez à calculer la taille des boutons en fonction de la fonte pour bien les disposer dans la fenêtre mère. Modifiez leur attribut de gravité pour qu'ils soient replacés automatiquement en cas de changement de taille.
2) Programmez le tracer des lignes dans la fenêtre graphique pour obtenir le comportement décrit plus haut. (On pourra s'inspirer fortement de l'exercice de tracer d'un rectangle avec la souris vu dans le TP 3).
3) En utilisant le Pixmap de sauvegarde du contenu de la fenêtre graphique, traitez les événements de type Expose sur cette fenêtre. (cf. la copie de la racine de l'exercice 5, TP1).
4) Programmez la réaction suivante au cliquer-relâcher de la souris sur les boutons (dans un premier temps, sans vous préoccuper de modifier l'apparence des boutons):
    a. Le bouton Hide efface le contenu de la fenêtre graphique (mais pas celui du Pixmap de sauvegarde).
    b. Le bouton Show rétablit le contenu de la fenêtre graphique (sauvegardé dans le Pixmap).
    c. Le bouton Clear efface le contenu de la fenêtre graphique et celui du Pixmap. (On repart à zéro).
5) Saisir les entrées au clavier pour que l'enfoncement des touches Control-C dans la fenêtre principale permette de quitter le programme.
6) Traitez le changement de taille de la fenêtre principale (on doit modifier la taille de la sous-fenêtre graphique).
7) Faites en sorte que le clic de souris sur une fenêtre bouton soit souligné par un affichage en "inverse-video" du bouton, et que le relâcher rétablisse ensuite l'affichage normal.
8) Veillez ensuite à ce que la commande associée au bouton ne soit exécutée que si l'on relâche le bouton de souris au dessus de la fenêtre bouton, et non à l'extérieur.

Institut Galilée
Année 2004-2005

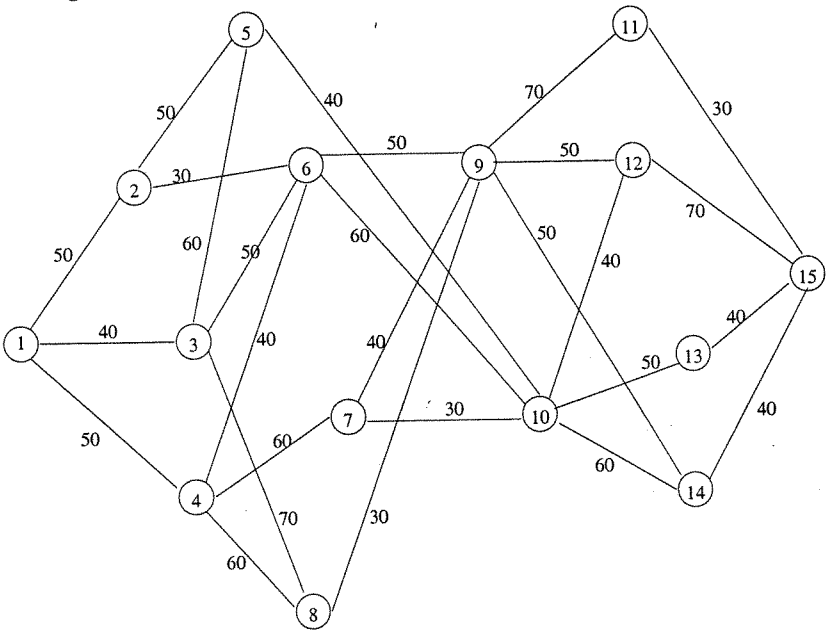
Théorie des graphes - Algo3
L3 Informatique - ILOG2

Examen

Vendredi 04 février 2005
Cours et TD autorisés - Durée : 3 heures

Exercice 1 : Tracé d'une autoroute

Quinze villes, numérotées de 1 à 15, sont reliées par des routes. La figure ci-dessous porte la longueur kilométrique de chaque tronçon.



Distances kilométriques

Pour chaque réponse aux questions suivantes, vous donnerez l'algorithme que vous utiliserez et vous détaillerez les différentes étapes de l'algorithme.

1. Donner un arbre couvrant de longueur minimale pour ce graphe.
2. Que représente un tel arbre pour le réseau routier considéré ?
3. Quel est le chemin le plus court permettant de joindre la ville 1 à la ville 15 ?
4. On a évalué, pour chaque route, le nombre horaire maximal de véhicules qu'elle peut écouler, compte tenu des ralentissements aux traversées des villes et villages, des arrêts aux passages à niveau, etc. Ces évaluations ont été les suivantes :

$(1,2)$	$\rightarrow$	750	$(2,5)$	$\rightarrow$	600	$(5,10)$	$\rightarrow$	650	$(9,11)$	$\rightarrow$	500	$(11,15)$	$\rightarrow$	600
$(1,3)$	$\rightarrow$	650	$(2,6)$	$\rightarrow$	400	$(6,9)$	$\rightarrow$	700	$(9,12)$	$\rightarrow$	150	$(12,15)$	$\rightarrow$	250
$(1,4)$	$\rightarrow$	800	$(3,5)$	$\rightarrow$	200	$(6,10)$	$\rightarrow$	200	$(9,14)$	$\rightarrow$	150	$(13,15)$	$\rightarrow$	750
			$(3,6)$	$\rightarrow$	350	$(7,9)$	$\rightarrow$	250	$(10,12)$	$\rightarrow$	350	$(14,15)$	$\rightarrow$	400
			$(3,8)$	$\rightarrow$	150	$(7,10)$	$\rightarrow$	150	$(10,13)$	$\rightarrow$	700			
			$(4,6)$	$\rightarrow$	300	$(8,9)$	$\rightarrow$	300	$(10,14)$	$\rightarrow$	250			
			$(4,7)$	$\rightarrow$	250									
			$(4,8)$	$\rightarrow$	200									

Quel est le débit horaire total maximal de véhicules susceptible de s'écouler entre les villes 1 et 15 ?

5. Afin d'accroître les possibilités de trafic routier entre les extrémités du réseau, un projet de construction d'une autoroute est étudié. Connaissant le coût des travaux pour chaque tronçon d'autoroute, quel sera le tracé le plus économique ?

Coûts des tronçons (en millions d'euros)

$(1,2)$	$\rightarrow$	1000	$(2,5)$	$\rightarrow$	1100	$(5,10)$	$\rightarrow$	900	$(9,11)$	$\rightarrow$	1100	$(11,15)$	$\rightarrow$	500
$(1,3)$	$\rightarrow$	600	$(2,6)$	$\rightarrow$	700	$(6,9)$	$\rightarrow$	1400	$(9,12)$	$\rightarrow$	1200	$(12,15)$	$\rightarrow$	700
$(1,4)$	$\rightarrow$	800	$(3,5)$	$\rightarrow$	1300	$(6,10)$	$\rightarrow$	1400	$(9,14)$	$\rightarrow$	900	$(13,15)$	$\rightarrow$	900
			$(3,6)$	$\rightarrow$	800	$(7,9)$	$\rightarrow$	1000	$(10,12)$	$\rightarrow$	600	$(14,15)$	$\rightarrow$	1000
			$(3,8)$	$\rightarrow$	1600	$(7,10)$	$\rightarrow$	800	$(10,13)$	$\rightarrow$	1100			
			$(4,6)$	$\rightarrow$	700	$(8,9)$	$\rightarrow$	800	$(10,14)$	$\rightarrow$	800			
			$(4,7)$	$\rightarrow$	1300									
			$(4,8)$	$\rightarrow$	700									

6. Parmi les cinq tronçons de l'autoroute, déterminées à la question précédente, dans quel ordre a-t-on intérêt à construire les deux premiers tronçons si l'on désire améliorer le trafic d'ensemble le plus vite possible ? L'autoroute en question permettra l'écoulement de 1500 véhicules à l'heure.

Exercice 2 : Composantes fortement connexes et plus courts chemins

1. Déterminer les composantes fortement connexes du 1-graphe orienté $G = (X, \Gamma)$ d'ordre 10 défini ci-dessous :

X	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
$\Gamma(X)$	x_6, x_7, x_8	x_3	x_7, x_{10}	x_1	x_3, x_8, x_9, x_{10}	x_4, x_7	$\emptyset$	x_4	x_5, x_7	x_2

Détailler les différentes étapes de l'algorithme de Kosaraju-Sharir ainsi que les différents types d'arcs obtenus par chaque application de DFS.
`Rappel: par convention, la construction de chaque arborescence des deux forêts couvrantes se fera en cherchant à atteindre prioritairement le sommet de plus petit numéro.`

2. En déduire le graphe réduit GR associé à G.

3. Appliquer l'algorithme de Dijkstra pour déterminer le chemin de valeur minimale issu de la source vers le puits de GR, connaissant les valeurs des arcs suivants :

Arcs	(x_5, x_{10})	(x_5, x_3)	(x_5, x_8)	(x_9, x_7)	(x_3, x_7)	(x_1, x_7)	(x_6, x_7)
Valeurs	4	6	5	7	4	2	1

Pour la valuation de GR, lorsqu'il y a le choix, sélectionner le meilleur arc de liaison entre deux composantes fortement connexes. Détailler dans un tableau les différentes étapes de l'algorithme de Dijkstra.

Université PARIS-NORD
Institut Galilée

2004-2005

Ingénierie Logicielle (seconde année)

Examen d'Analyse Numérique
Jeudi 3 Février 2005 (9h à 11h)

I

On considère la matrice carrée A d'ordre n dont tous les coefficients sont donnés par les relations suivantes :

$a_{i,i+1} = \alpha_i \qquad 1 \le i \le n-1$

$a_{i,i-1} = \alpha_{i-1} \qquad 2 \le i \le n$

$a_{n,n} = \alpha_{n-1}$

$a_{i,j} = 0$ dans les autres cas.

De plus: $\alpha_i \neq 0 \quad \forall i \; (i = 1,2,3,\ldots\ldots, n-1)$

1) Ecrire les n équations du système linéaire : $A\,x = b$, où $b = (b_1, b_2, \ldots\ldots, b_n)$ est un vecteur de R^n (On explicitera les 3 premières équations, les 3 dernières et l'équation générale d'indice i).

2) Donner la solution explicite du système si n est pair , puis si n est impair (On remplacera dans les formules n par 2k puis par 2k+1).

3) Trouver une formulation identique dans le cas pair et impair.

4) Comment peut-on minimiser l'encombrement mémoire ?

II

1) Vérifier que la méthode de Gauss-Seidel est consistante.

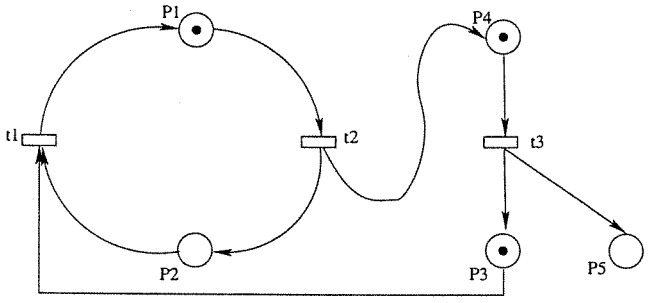
2) Expliciter les formules qui permettent de calculer les composantes de $x^{(m+1)}$ connaissant celles de $x^{(m)}$.

3) Écrire en Matlab la résolution du système linéaire $A\,x = b$ (A régulière), de dimension n, par la méthode de Gauss-Seidel, sous la forme d'une fonction de paramètres A, b et n.

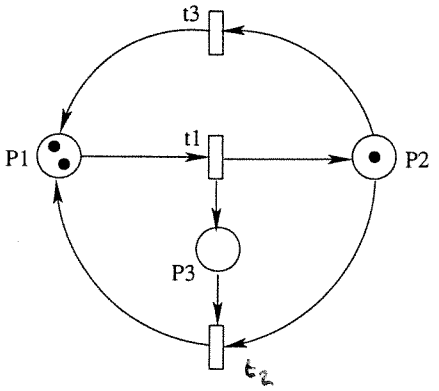
Examen de Génie Logiciel - GLIII

CC - Janvier 2005

L'épreuve dure 2 heures. Les notes de cours et de TD manuscrites ainsi que les énoncés et supports de cours distribués cette année sont les seuls documents autorisés. Les téléphones portables doivent être éteints. Soignez l'écriture et la présentation (cela peut rapporter 1 à 2 points), écrivez avec un stylo d'encre suffisamment sombre.



1. (a) Donner, pour le réseau ci-dessus, les matrices Pre, Post, ainsi que la matrice d'incidence.
   (b) Partant du marquage initial donné, donner le marquage obtenu après la séquence de transitions < t2, t1, t3, t3 >.



2. Décrire le fonctionnement du réseau ci-dessus et calculer en détaillant les premières étapes de calcul
   (a) le graphe des marquages accessibles
   (b) l'arborescence de couverture
   (c) le graphe de couverture.
   (d) Quelles sont les propriétés de ce réseau ?
3. Modélisez à l'aide d'un réseau de Petri le fonctionnement d'un cabinet de consultation d'un médecin qui ne permet l'attente que d'un patient à la fois.
4. Pour le système décrit ci-après :
   (a) Identifier quels sont les éléments pertinents de ce système et développez les cas d'utilisation pour ce problème.
   (b) Donner une description détaillée de l'emprunt d'un livre.
   (c) Indiquez ensuite de quel type de système il s'agit, et développez les éléments de spécification correspondants. Donnez les diagrammes correspondant à la méthode vue en cours.

   Le but de cet exercice est de concevoir un système automatisé d'emprunt de livres de bibliothèque (LBBS = *Library Book Borrowing System*) pouvant être utilisé dans les bibliothèques de département des universités. LBBS permettrait d'abaisser la charge de travail des bibliothécaires en automatisant le traitement des emprunts de livre. Le système ne nécessite pas que les utilisateurs s'identifient avant de rechercher des ouvrages selon certains critères, ou de vérifier la disponibilité d'un livre en particulier. Cependant, pour

emprunter des livres, pour vérifier l'état de ses emprunts ou pour réserver des livres déjà empruntés, les utilisateurs doivent au préalable s'identifier auprès du système.

Lors de chaque session, un reçu est imprimé, détaillant pour chaque livre emprunté :

- le titre du livre,
- l'identifiant unique correspondant au livre,
- la date d'emprunt du livre,
- et la date à laquelle le livre doit être rendu.

Au début de chaque semaine, le système envoie un e-mail d'avertissement à chaque personne ayant emprunté un ouvrage et ayant dépassé la période de prêt autorisé.

Le serveur central de l'université centralise toutes les informations concernant les utilisateurs du système (catégorie, adresse e-mail, ...etc.).

Un utilisateur s'identifie auprès du système à l'aide d'une carte et d'un mot de passe (comme avec les distributeurs automatiques de billets). Pour des raisons de sécurité, le serveur central de l'université se charge de réaliser toutes les vérifications de mot de passe. Pour cela, le système bénéficie d'une connexion Ethernet à ce serveur.

Des codes barres sont apposés aux livres pour pouvoir les identifier lors d'un emprunt (un lecteur de codes barres est à la disposition des utilisateurs). Dans le cas où un code barre est endommagé, rendant sa lecture impossible, il est possible de rentrer manuellement sa valeur.

Chaque bibliothèque de département possède son propre système de recherche de livres, que LBBS doit utiliser lors de toutes les recherches d'ouvrage. Ces systèmes ne peuvent pas être modifiés car il s'agit de logiciels propriétaires dont le code n'est pas disponible. Lorsqu'un seul de ces systèmes est en cours de recherche, la recherche d'ouvrage n'est plus disponible auprès des autres utilisateurs. Toutes les recherches doivent s'effectuer *via* LBBS.

Le renouvellement d'emprunt (*i.e.* l'extention de la date de retour) n'est pas pris en charge dans cette version du système, même si cette caractéristique est prévue pour une future version. Il s'agit d'une caractéristique important devant être supportée *via* le web.

Voici les différentes règles d'emprunt :

- Chaque utilisateur peut emprunter un nombre limité de livres en même temps, en fonction de la catégorie de l'utilisateur. Par exemple, les directeurs de département peuvent emprunter jusqu'à 100 ouvrages simultanément alors que les assistant sont limités à 15.
- Pour chaque catégorie de livre, la catégorie d'utilisateur définit aussi la période d'emprunt d'un ouvrage. Par exemple, un directeur de département peut emprunter un livre standard durant 6 mois tandis qu'un assistant est limité à 3 mois.
- La période durant laquelle un utilisateur peut emprunter un ouvrage dépend de la catégorie de ce livre. Par exemple, un assistant peut emprunter un livre standard durant 3 mois alors que l'emprunt d'un périodique sera limité à 1 mois. De plus, certains livres ne peuvent pas être empruntés par certaines catégories d'utilisateur. *E.g.* les assistants et les étudiants ne peuvent pas emprunter les livres de référence.

Université Paris-Nord
Institut Galilée

ILOG2
Année 2004-2005

Partiel 1 d'algorithmique randonisée (Algo III.2)

Jeudi 7 avril 2005 de 10 heures à midi

Avertissement. Les documents et les calculatrices sont interdits.
SOIGNEZ LA RÉDACTION. RESPECTEZ LES NOTATIONS DE L'ÉNONCÉ.

Exercice 1 (application de la méthode probabiliste)

K_n désigne la clique à n sommets, c'est-à-dire le graphe complet à n sommets.

Soit $\ell < n$. Montrez qu'il existe un 2-coloriage des arêtes de la clique K_n qui laisse au plus

$$\binom{n}{\ell} 2^{1-\binom{\ell}{2}}$$

sous-cliques K_ℓ monochromes.

Exercice 2 (calcul d'espérance et de variance)

Soit σ une permutation de $\{1, \ldots, n\}$. Un couple (i, j), $1 \leqslant i < j \leqslant n$, est une *inversion* de σ si $\sigma(j) < \sigma(i)$. On désigne par $\mathrm{Inv}(\sigma)$ le *nombre d'inversions* de σ. Dans la suite, on considérera que toutes les permutations de $\{1, \ldots, n\}$ sont équiprobables.

a) Montrez que $\mathbf{E}[\mathrm{Inv}(\sigma)] = \dfrac{n(n-1)}{4}$. [Indication : utilisez I_{ij}, variable aléatoire indicatrice de l'événement "(i, j) est une inversion".]

b) Montrez que $\mathbf{Var}(\mathrm{Inv}(\sigma)) = \dfrac{n(n-1)(2n+5)}{72}$. [Indication : utilisez la variable aléatoire $N_j = \sum_{i<j} I_{ij}$.]

c) On considère le tri par insertion d'une liste de n valeurs deux à deux distinctes. Par exemple, si la liste de départ est

6 1 4 2 5 3

les états après chaque insertion sont (on a fait apparaître en gras les valeurs qui sont déplacées au cours de l'insertion) :

6
1 **6**
1 4 **6**
1 2 **4** **6**
1 2 4 5 **6**
1 2 3 **4** **5** **6**

On désigne par X le nombre total de valeurs déplacées au cours du tri. En supposant que tous les ordres initiaux sont équiprobables, calculez $\mathbf{E}[X]$ et $\mathbf{Var}(X)$.

Exercice 3 (analyse d'algorithme randonisé)

On considère l'algorithme randonisé récursif suivant, Select(S, k), qui sélectionne le k^{e} élément d'un ensemble S d'au moins k entiers deux à deux distincts :

Select(S, k) :

1. Tirer au hasard un élément $x \in S$;
2. En comparant x à chaque élément de $S - \{x\}$, constituer les ensembles
   $S_- := \{y \in S \mid y < x\}$ et
   $S_+ := \{y \in S \mid y > x\}$;
3. Si $|S_-| = k - 1$, retourner x ;
   Si $|S_-| > k - 1$, retourner Select(S_-, k) ;
   Si $|S_-| < k - 1$, retourner Select$(S_+, k - |S_-| - 1)$.

a) En supposant qu'à l'étape 1, tous les éléments de S sont équiprobables, calculez le nombre moyen de comparaisons lorsque l'algorithme Select(S, k) est exécuté sur un ensemble S contenant n éléments deux à deux distincts. [Indication : utilisez X_{ij}, variable aléatoire indicatrice de l'événement "le i^{e} et le j^{e} éléments sont comparés au cours de l'exécution", et distinguez trois cas, selon les positions respectives de i et j par rapport à k.]

b) Donnez une estimation de la valeur calculée en 3.*a* quand n, $k \longrightarrow \infty$ avec $k = \alpha n$, où $0 < \alpha < 1$. [Rappel : $\lim_{n\to\infty} \left\{ \sum_{i=1}^{n} \frac{1}{i} - \ln n \right\} = \gamma$, où γ est la constante d'Euler : $\gamma \approx 0,57721$.]

Université Paris-Nord
Institut Galilée

ILOG2
Année 2004-2005

Partiel 2 d'algorithmique randonisée (Algo III.2)

Vendredi 10 juin 2005 de 10 heures à midi

Avertissement. Les notes de cours et de TD sont autorisées. Les calculatrices sont interdites.
SOIGNEZ LA RÉDACTION. RESPECTEZ LES NOTATIONS DE L'ÉNONCÉ.

Exercice 1 (calcul d'espérances)

Dans cet exercice, σ désigne une permutation de l'ensemble $\{1, \ldots, n\}$. On suppose que toutes les permutations de $\{1, \ldots, n\}$ sont équiprobables.

a) On dit que $i \in \{1, \ldots, n\}$ est un *point fixe* de σ si $\sigma(i) = i$.
Calculez $\mathbf{E}[F]$, où F est le nombre de points fixes de σ.

On rappelle que $\sigma(i_1), \ldots, \sigma(i_k)$ est une *sous-suite croissante* de σ si et seulement si $i_1 < i_2 < \cdots < i_k$ et $\sigma(i_1) < \sigma(i_2) < \cdots < \sigma(i_k)$.

b) Une *sous-suite croissante maximale* est une sous-suite croissante qui n'est incluse dans aucune autre sous-suite croissante. Par exemple, la permutation

$$3,\ 5,\ 1,\ 2,\ 4,\ 6$$

a exactement deux sous-suites croissantes maximales : $3, 5, 6$ et $1, 2, 4, 6$.

Calculez $\mathbf{E}[M]$, où M est le nombre de sous-suites croissantes maximales de σ. [Indication. Considérez I_j, variable aléatoire indicatrice de l'événement "Il existe une sous-suite croissante maximale qui commence par la valeur j."]

c) Une *suite montante* est une sous-suite croissante de valeurs consécutives qui n'est incluse dans aucune autre sous-suite croissante de valeurs consécutives. Par exemple, la permutation

$$4,\ 3,\ 5,\ 1,\ 2,\ 6$$

a exactement trois sous-suites montantes : $4, 5, 6$; 3 et $1, 2$.

Calculez $\mathbf{E}[R]$, où R est le nombre de sous-suites montantes de σ.

d) Calculez $\mathbf{E}[X]$, où X est le nombre de sous-suites croissantes de σ.

Exercice 2 (Algorithme Las Vegas pour une grande coupe)

On cherche une coupe aussi grande que possible dans un graphe non orienté (sans boucle).

a) Soit $G = (X, U)$ un graphe non orienté sans boucle, ayant n sommets et m arêtes. Montrez, en utilisant la méthode probabiliste, qu'il existe une partition de $X = A \cup B$ qui définit une coupe de taille $m/2$.

b) En utilisant la question précédente, montrez qu'il existe un algorithme Las Vegas de complexité polynomiale (en le nombre de sommets) qui calcule, dans un graphe non orienté sans boucle, une coupe de taille au moins égale à la moitié du nombre d'arêtes.

Exercice 3 (Marche aléatoire : la ruine du joueur)

On considère deux joueurs A et B qui, à chaque tour (identifié à un instant du processus), misent chacun 1 euro. Le joueur qui remporte le tour rafle la mise et on suppose que les tours sont indépendants et qu'à chaque tour, les deux joueurs ont chacun une chance sur deux de gagner. Ce jeu est représenté par un système dont l'état à l'instant t est le nombre d'euros "gagnés" par le joueur A depuis le début du jeu ; à l'instant $t = 0$, l'état du système du système vaut donc 0. Il est raisonnable de supposer que le joueur A ne peut pas perdre plus de ℓ_A euros et que le joueur B ne peut pas perdre plus de ℓ_B euros : le jeu cesse donc quand il atteint l'état $-\ell_A$ ou l'état ℓ_B (qui correspondent l'un et l'autre à la ruine d'un des joueurs).

a) Montrez comment modéliser ce jeu à l'aide d'une chaîne de Markov. Identifiez les états récurrents et transitoires.

b) Soit j un entier satisfaisant $-\ell_A \leqslant j \leqslant \ell_B$. On désigne par q_j la probabilité d'aboutir à la ruine du joueur B à partir de l'état j. Écrivez une relation de récurrence liant q_j, q_{j-1} et q_{j+1}, pour $-\ell_A < j < \ell_B$.

c) Résolvez l'équation de récurrence de la question *b*. Déduisez-en la probabilité que le joueur B soit ruiné par le jeu.

Institut Galilée - Université Paris 13
ILOG2

OC2
08 avril 2005

Partiel
Durée : 3 heures
Notes de cours et de TD autorisés

Exercice 1

On envisage le problème d'affectation (P) suivant :
$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$
$s.c. \sum_{j=1}^{n} x_{ij} = 1 \quad \forall i = 1, \ldots, n \quad (1)$
$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j = 1, \ldots, n \quad (2)$
$x_{ij} \in \{0, 1\} \quad \forall i, j = 1, \ldots, n \quad (3)$
qui consiste à affecter n tâches $j \in \{1, \ldots, n\}$ à n machines $i \in \{1, \ldots, n\}$ au moindre coût de réalisation des tâches (tous les c_{ij} sont supposés positifs).

1. Démontrer pourquoi les contraintes (3) peuvent être remplacées par
   $x_{ij} \geq 0 \quad \forall i, j = 1, \ldots, n \quad (3')$
2. En déduire que le problème (P) est un cas particulier du problème de transport.
3. Résoudre par l'algorithme primal-dual le problème (P) dont les données sont les suivantes (4 machines et 4 tâches) :

10	8	6	4
10	8	16	2
8	2	8	0
2	0	8	0

Préciser sa valeur et la solution optimale et réalisable associée.
(`Utiliser l'heuristique initiale des chemins améliorants ''1ère usine non satisfaite vers 1er client non satisfait''.`)

Exercice 2

On considère le programme linéaire en variables entières (P) suivant :
$min\ 5x_1 + 3x_2$
$s.c.\ 4x_1 - x_2 \leq 8$
$-x_1 + x_2 \leq 1$
$3x_1 + 2x_2 \geq 6$
$x_1 \geq 0,\ x_2 \geq 0$ et entiers
et le problème en variables continues associé noté (PL).

1. Faire les résolutions graphiques de (P) et de (PL). Préciser les valeurs des problèmes et les solutions optimales x^* de (P) et x^{PL} de (PL). Dessiner également le domaine $Conv(Dom(P))$.
2. Quelle est la valeur atteinte au point x^{heur} tel que $x_j^{heur} = \lceil x_j^{PL} \rceil$, $j = 1, 2$? Quel est l'écart entre $\lceil v(PL) \rceil$ et cette valeur ?
3. Quel aurait été cet écart si l'objectif de (P) avait été $f = (1, 0)$?

Exercice 3

Rédiger une version non récursive de l'algorithme de complexité linéaire, recherchant les k plus grands éléments d'un ensemble de n réels distincts t_j $j = 1, \ldots, n$ (avec $k \leq n$).
(Il est demandé d'utiliser le principe et les notations (E_1, h, E_2) de l'algorithme de complexité linéaire du cours, résolvant le programme linéaire $(\overline{K})$ associé à un sac à dos $0-1$ (K).)

Exercice 4

Appliquer l'algorithme de complexité linéaire résolvant le programme linéaire $(\overline{K})$ associé au sac à dos $0-1$ (K) suivant :
$max\ 5x_1 + 8x_2 + 19x_3 + 14x_4 + 9x_5 + 3x_6 + 20x_7$
$s.c.\ 4x_1 + 4x_2 + 20x_3 + 15x_4 + 13x_5 + 2x_6 + 20x_7 \leq 23$
$x_j \in \{0,1\}\ j = 1, \ldots, 7$

Exercice 5

Etant donné le problème du sac-à-dos (K) à six variables bivalentes :
$max\ 30x_1 + 25x_2 + 22x_3 + 40x_4 + 28x_5 + 20x_6$
$s.c.\ x_1 + 4x_2 + 7x_3 + 16x_4 + 13x_5 + 10x_6 \leq 26$
$x_j \in \{0,1\}\ j = 1, \ldots, 6$

1.

Résoudre le programme linéaire en variables continues $(\overline{K})$ associé à (K).

2.

Déterminer une solution réalisable de (K) par l'heuristique gloutonne et la valeur $\underline{v}(K)$ associée.

3.

Calculer les coûts réduits optimaux de $(\overline{K})$; en déduire que pour espérer trouver une valeur supérieure à $\underline{v}(K)$, les variables x_1 et x_2 doivent être fixées à la valeur 1.

4.

Première façon de poursuivre la résolution de (K).

4.1

Utiliser la relaxation lagrangienne associée à $u = 2,1$ pour fixer la variable x_3 à la valeur 1.

4.2

Déduire par implication logique la fixation à 0 d'une nouvelle variable.

4.3

Résoudre le problème ainsi réduit à deux variables ; en déduire la solution optimale ainsi que la valeur de (K).

5

Autre façon de poursuivre la résolution de (K) (i.e. autre suite de la question 3).

5.1

On note (K') le problème $(K|x_1 = x_2 = 1)$.
Démontrer que la variable x_4 doit être fixée à la valeur 0 en déterminant la solution évidente en variables 0-1 du problème $(K'|x_4 = 1)$

5.2

On note (K'') le problème $(K|x_1 = x_2 = 1; x_4 = 0)$.
Résoudre le programme linéaire en variables continues $(\overline{K''})$ associé à (K'').

5.3

Calculer les coûts réduits optimaux de $(\overline{K''})$; en déduire que pour espérer trouver une valeur supérieure à $\underline{v}(K)$, les variables x_3 et x_5 doivent être fixées à la valeur 1.

5.4

Retrouver par implication logique la solution optimale ainsi que la valeur de (K).

Institut Galilée - Université Paris 13
ILOG2

OC2
17 juin 2005

Examen
Durée : 3 heures
Notes de cours et de TD autorisées

Exercice 1

Etant donné le problème du sac-à-dos (K) à quatre variables bivalentes :
$max\ 14x_1 + 4x_2 + 3x_3 + 8x_4$
$s.c.\ 15x_1 + 4x_2 + 2x_3 + 4x_4 \leq 23$
$x_j \in \{0,1\}\ j = 1,\ldots,4$

1.

Résoudre (K) par programmation dynamique en gérant des ensembles de triplets $\left(\sum_{j\in J} a_j, \sum_{j\in J} c_j, J\right)$ non dominés pour déterminer la valeur de (K) et sa solution optimale et réalisable.

2.

Quelle aurait été la valeur de (K) si le second membre avait été égal à 20 ?

3.

Appliquer l'algorithme glouton dans l'ordre décroissant des $\frac{c_j}{a_j}$, $j = 1,\ldots,4$, pour déterminer une solution réalisable $\underline{x}$ de (K).

4.

Appliquer une recherche de type Tabou avec $\underline{x}$ pour solution initiale jusqu'à obtenir la valeur $v(K)$ trouvée à la question 1.

Exercice 2

On envisage le problème du sac-à-dos (K) à sept variables bivalentes :
$max\ 30x_1 + 20x_2 + 60x_3 + 10x_4 + 8x_5 + 2x_6 + x_7$
$s.c.\ 5x_1 + 5x_2 + 20x_3 + 5x_4 + 4x_5 + 2x_6 + 2x_7 \leq 37$
$x_j \in \{0,1\}\ j = 1,\ldots,7$

1.

Appliquer l'algorithme glouton pour déterminer une solution réalisable $\underline{x}$ de (K) et le minorant $\underline{v}(K)$ associé de $v(K)$.

2.

Ecrire la relaxation lagrangienne de (K) pour le multiplicateur $u = \frac{c_6}{a_6} = 1$.

3.

Utiliser cette relaxation lagrangienne pour fixer 5 variables de (K) à la valeur 1.

4.

En déduire la valeur et la solution optimale et réalisable de (K). Expliquer.

Exercice 3

On envisage le programme linéaire en nombres entiers (P) suivant :

$$\begin{gathered}\min 5x_1 + 3x_2 \\ \text{s.c. } 4x_1 - x_2 \leq 8 \ (1) \\ -x_1 + x_2 \leq 1 \ (2) \\ 3x_1 + 2x_2 \geq 6 \ (3) \\ x_1, x_2 \in \mathbb{N}\end{gathered}$$

dont on suppose connu le tableau optimal du programme linéaire en variables continues associé (PL) :

Base	t	T^1	T^2	T^3	T^4	T^5
A^3	33/5	0	0	1	11/5	3/5
A^1	4/5	1	0	0	−2/5	−1/5
A^2	9/5	0	1	0	3/5	−1/5
	47/5	0	0	0	−1/5	−13/5

où x_3, x_4 et x_5 sont les variables d'écart ou de surplus respectives des contraintes (1), (2) et (3).

1.

Rappeler les résolutions graphiques de (P) et de (PL).
[Dessiner également le domaine Conv(Dom(P)) ; préciser les valeurs des problèmes et les solutions optimales x^* de (P) et x^{PL} de (PL)].

2.

Déterminer et représenter graphiquement
- la coupe de Dantzig
- les coupes de Gomory associées à x_1, à x_2 et à x_3, à chaque fois pour $h = 1$.

3.

Parmi ces quatre coupes, laquelle permettrait de résoudre la problème (P) par la résolution en continu ? Expliquer.

4.

Résoudre (P) par un algorithme de branch-and-bound en n'utilisant que les résolutions graphiques des programmes linéaires en continu associés aux sous-problèmes de chaque nœud de l'arborescence.

Exercice 4

On considère le programme linéaire en nombres entiers (P) suivant :
$max\ x_1 + x_2$
$s.c.\ 9x_1 + 3x_2 \leq 20 \quad (1)$
$3x_1 + 6x_2 \leq 14 \quad (2)$
$x_1,\ x_2 \geq 0$ et entiers $\quad (3)$
et le problème en variables continues associé noté (PL).

1.

Déterminer successivement par résolution graphique
- les valeurs et solutions optimales et réalisables de (P) et de (PL),

ainsi que les valeurs
- du dual lagrangien (DRL^1) obtenu en dualisant la contrainte (1),
- du dual lagrangien (DRL^2) obtenu en dualisant la contrainte (2),
- du dual (DDL) de la décomposition lagrangienne
- du dual de la relaxation agrégée (DRA), combinant les contraintes (1) et (2).

2.

a.

Quelles sont les coupes de Chvatal associées respectivement aux contraintes (1) et (2) ?

b.

Quel domaine de la question 1 retrouve-t-on lorsque ces deux coupes remplacent les contraintes (1) et (2) ?

c.

Agréger ces deux coupes avec les poids 1. Déterminer la coupe de Chvatal de cette nouvelle contrainte avec $u = 1/3$. Quel domaine de la question 1 retrouve-t-on ainsi ?

Exercice 5

Soit le problème en nombres entiers (P) suivant :

$$\begin{gathered} \max x_1 + x_2 \\ \text{s.c. } 2x_1 + 5x_2 \leq 10 \\ 4x_1 - 3x_2 \leq 6 \\ x_1, x_2 \in \mathbb{N} \end{gathered}$$

(traité en cours par l'algorithme de Gomory).

Vérifier que les deux premières itérations de l'algorithme primal du simplexe en nombres entiers permettent de détecter (sans le prouver) la valeur de (P) et sa solution optimale et réalisable ; représenter graphiquement les deux coupes de Gomory engendrées.
(`Il est demandé de démarrer l'algorithme en choisissant` $s = 1$.)

Université Paris 13
Institut *Galilée*

ILOG 2 & MI1
2004-2005

Algorithmique parallèle et distribuée
Examen du mardi 25 janvier 2005

(*Documents autorisés : notes de cours et de T.D.*)

1 Plus long préfixe

Soit une chaîne de caractères C comportant n lettres. Le problème consiste à déterminer le plus grand entier k tel que

$$C[1]C[2]\dots C[k] = C[n-k+1]\dots C[n-1]C[n].$$

Autrement dit, k est la taille du plus long préfixe de C qui est aussi un suffixe de C.

Donner un algorithme parallèle *CRCW* de complexité en temps $T(n) = \mathcal{O}(1)$. Cet algorithme est-il efficace ? Optimal ?

2 Calcul parallèle de $n!$

Le but est de donner un algorithme parallèle qui calcule $n!$, la factorielle d'un entier donné n, en temps $\mathcal{O}(\log(n))$ avec $\mathcal{O}(n)$ processeurs.

2.1 Algorithme de la factorielle

a) Spécifier l'algorithme qui calcule $n!$ sous ces hypothèses.

b) Préciser la technique ainsi que le type d'algorithme parallèle (de PRAM) utilisé.

c) L'algorithme obtenu est-il efficace ? Optimal ?

3 Produit de deux matrices

Soit un algorithme parallèle *CRCW* calculant le produit d'une matrice A $(m \times k)$ et d'une matrice B $(\ell \times n)$ selon la formule classique du produit de deux matrices quelconques.

3.1 Formule du produit de deux matrices

Sous quelles conditions le produit $A \times B$ existe-t-il ? Rappeler la formule calculant les coefficients de la matrice produit C $(m \times n)$.

3.2 Algorithme produit *CRCW* « optimal »

Préciser comment résoudre les *conflits en lecture/écriture* pour obtenir un algorithme fondé sur la formule de 3.1 ci-dessus *et* « optimal » *par rapport* à l'algorithme séquentiel correspondant, de complexité en temps $T_s(n) = \mathcal{O}(n^3)$ ($n = \max\{n, k, m\}$).

3.3 Procédure *ProduitMatrices*(*A*,*B*,*C*)

(i) Écrire une procédure *CRCW ProduitMatrices*(A,B,C), « optimale » *au sens* du 3.2.

(ii) Calculer la complexité $T_{CRCW}(n)$ de *ProduitMatrices*(A,B,C). En déduire que la procédure est bien « optimale » (toujours *au sens* du 3.2).

3.4 Algorithme produit *EREW*

a) Sous les hypothèses du 3.3, peut-on calculer le produit $A \times B$ avec un algorithme *EREW* ? Ses performances sont-elles alors les mêmes ?

b) Décrire un algorithme *EREW* de complexité en temps $T_{EREW}(n) = \mathcal{O}(\log(n))$ calculant $A \times B$ (matrices carrées $n \times n$) à l'aide de n^3 processeurs. L'algorithme *EREW* obtenu est-il efficace ? Optimal ?

4 Recherche des racines d'un arbre binaire

Considérons une forêt $\mathcal{F}$ d'arbres binaires et l'algorithme qui permet à chaque nœud de connaître la racine de son arbre. On se restreint ici au cas particulier où $\mathcal{F}$ n'est constituée que d'*un seul* arbre binaire à n nœuds.

a) Écrire un algorithme *TrouverRacines* qui résout le problème.

b) Montrer qu'avec cette hypothèse, on peut faire en sorte qu'un algorithme *CRCW TrouverRacines* s'exécute en $\mathcal{O}(1)$, indépendamment de la profondeur de l'arbre.

c) Montrer qu'une implémentation *EREW* quelconque de *TrouverRacines* nécessite un temps $\Omega(\log(n))$.

d) Les algorithmes *CRCW* et *EREW* obtenus respectivement en b) et c) sont-il efficaces ? Optimaux ?

5 Classement optimal d'une liste chaînée

Soit une liste chaînée chaînée L contenant n objets. L'algorithme *EREW* de classement par sauts de pointeurs vu en cours est efficace mais *non optimal.*

Donner un algorithme *ClassementListe*(L) *optimal*, de complexité en temps $T(n) = \mathcal{O}(\log(n))$ et utilisant $\mathcal{O}(n/\log(n))$ processeurs comme suit.

i) Réduire la liste L à une liste L' de taille $\mathcal{O}(n/\log(n))$.

ii) Utiliser l'algorithme habituel de classement sur la liste réduite L', puis récupérer les objets « ôtés » lors de l'initialisation et calculer leur rang. Quel est le travail obtenu ? En déduire que *ClassementListe*(L) est bien optimal.

iii) Quelles hypothèses faut-il formuler sur les entrées de *ClassementListe*(L) pour opérer ainsi ?

6 Algorithme distribué d'élection sur un anneau

Soit un système distribué S représenté par un graphe fini, connexe, simple et symétrique $G = (X, U)$, où X constitue l'ensemble des processus de S et $U \subseteq X^2$ l'ensemble de ses lignes de communication ; on note $|X| = n$ et $|U| = m$.

L'algorithme distribué $\mathcal{A}$ spécifié ici (page 5) résout le problème de l'élection par comparaisons et calcul d'une identité *extrémale* sur un anneau *unidirectionnel* et *asynchrone*.

6.1 Hypothèses sur $\mathcal{A}$

Un certain nombre d'hypothèses simplificatrices sont faites ici sur $\mathcal{A}$, sans perte de généralité.

- La communication entre les n processus s'effectue par messages et *sans déséquencement* (communication *FIFO*).
- Le réseau en anneau est *asynchrone à délais bornés*.
- Chacun des n processus de X possède une identité unique, distincte de celle des autres processus et qu'il est seul à connaître.
  Ces identités appartiennent à l'ensemble fini $[n] = \{1, \ldots, n\} \subset (\mathbb{N}, <)$, l'ordre total des entiers.
- L'algorithme $\mathcal{A}$ est déclenché par un ensemble non vide $I \subseteq [n]$ d'identités de processus appelés *initiateurs*, de manière *indépendante* et *non simultanée*.

6.2 Preuve informelle de $\mathcal{A}$

Sous l'hypothèse d'un nombre quelconque $|I|$ d'initiateurs ($1 \leq |I| \leq n$), expliquer le déroulement global de l'algorithme $\mathcal{A}$.

a) Préciser le comportement des $|I|$ initiateurs et celui des $n - |I|$ processus non initiateurs (non spécifié dans l'algorithme donné page 5). Que fait un processus *passif*, *actif* ?

b) À partir de l'algorithme de Franklin vu en cours, montrer que $\mathcal{A}$ utilise le même principe de comparaison entre identités sous des hypothèses plus restrictives. Quelle est l'identité élue ici par l'algorithme $\mathcal{A}$?

c) L'algorithme $\mathcal{A}$ s'exécute à l'aide de trois identités quelconques, r, q et p, prises *dans cet ordre* sur l'anneau unidirectionnel. Autrement dit, q est l'identité du *premier prédécesseur actif* de p et r joue le même rôle pour q (compte tenu du sens de rotation des messages sur l'anneau). $\mathcal{A}$ est spécifié ici (page 5) pour l'identité p.

- Expliquer le rôle des variables $idcour_p$ (identité *courante* ou *provisoire*) et $idpred_p$ (identité du premier voisin actif de p qui le précède sur l'anneau) utilisées par p.
- Que se passe-t-il lorsque $idcour_p = idpred_p$?
- Montrer qu'une identité qui suit un minimum local n'est pas minimale. En déduire que si une phase commence avec $k > 1$ processus actifs d'identités *idcour* distinctes, alors le nombre de candidats restant en fin de phase est compris entre 1 et $k/2$.
- En déduire que $\mathcal{A}$ termine bien avec un unique élu.

6.3 Analyse de complexité de $\mathcal{A}$

Dans les calculs de complexité de l'algorithme $\mathcal{A}$ qui suivent, on fait l'hypothèse que *tous* les processus sont initiateurs : $|I| = n$ (sans perte de généralité).

6.3.1 Complexité en temps de $\mathcal{A}$ (cas extrêmes)

(i) Donner une configuration initiale de $\mathcal{A}$ qui requiert un nombre maximal de phases (virtuelles) : complexité en temps dans le *pire des cas.*

(ii) Montrer que le nombre maximal de phases $T_{\mathcal{A}}(n)$ de $\mathcal{A}$ est alors (*au pire*)

$$T_{max}(n) = \lfloor \lg(n) \rfloor + 1 = \lceil \lg(n+1) \rceil.$$

(iii) Donner une configuration initiale de $\mathcal{A}$ qui requiert seulement *deux phases*, quel que soit le nombre d'initiateurs. L'algorithme $\mathcal{A}$ peut-il se terminer en une seule phase ?

6.3.2 Complexité en communication de $\mathcal{A}$ (cas extrêmes)

(i) Calculer la complexité en messages *au pire* de $\mathcal{A}$.

- Montrer que le nombre de messages de tous types reçus par chaque processus est exactement de *deux* à chaque phase.
- En conclure que

$$C_{max}(n) \;=\; 2n \lg(n) + \mathcal{O}(n) \quad (n \to \infty).$$

(ii) Même question pour la complexité en messages de $\mathcal{A}$, $C_{min}(n)$, dans le *meilleur cas* (s'il existe).

6.3.3 Complexité en espace-mémoire de $\mathcal{A}$

Calculer la complexité en espace-mémoire $S(n)$ de $\mathcal{A}$.

6.3.4 Comparaisons de performances

Comparer les mesures de complexité respectives de $\mathcal{A}$ (questions 6.3.1, 6.3.2 et 6.3.3 ci-dessus) et celles des algorithmes de Franklin et de Chang-Roberts. Qu'en concluez-vous ?

ALGORITHME $\mathcal{A}$ (pour un processus d'identité p)

ÉTAPE 1 – ***Élection***

var $état_p$: {*élu,battu,actif,passif*} (**Init** *nondéf*) ;
$idcour_p$ (**Init** id_p), $idpred_p$ (**Init** *nondéf*), $vainq_p$ (**Init** *nondéf*) : $[n]$;

début **si** $p \in I$ **alors** $état_p := actif$ **sinon** $état_p := passif$ **fsi** ;

tantque $vainq_p = nondéf$ **faire**
si $état_p = actif$ **alors**
envoyer $\langle \mathbf{1}, idcour_p \rangle$; recevoir $\langle \mathbf{1}, q \rangle$; $idpred_p := q$;
si $idpred_p = idcour_p$ **alors** /* $idpred_p$ est un minimum local */
envoyer $\langle \mathbf{min}, idpred_p \rangle$; $vainq_p := idpred_p$;
recevoir $\langle \mathbf{min}, q \rangle$;
sinon /* $idpred_p$ est l'identité courante du prédécesseur actif de p */
envoyer $\langle \mathbf{2}, idpred_p \rangle$; recevoir $\langle \mathbf{2}, q \rangle$;
si $idpred_p < idcour_p$ **et** $idpred_p < q$
alors $idcour_p := idpred_p$
sinon $état_p := passif$
fsi
fsi
sinon /* $état_p = passif$ */
recevoir $\langle \mathbf{1}, q \rangle$; envoyer $\langle \mathbf{1}, q \rangle$; recevoir M ; envoyer M ;
/* $M = \langle \mathbf{min}, q \rangle$ ou bien $M = \langle \mathbf{2}, q \rangle$ */
si $M = \langle \mathbf{min}, q \rangle$ **alors** $vainq_p := q$ **fsi**
fsi
fintantque
si $id_i = vainq_i$ **alors** $état_p := élu$ **sinon** $état_p := battu$ **fsi**
fin

ÉTAPE 2 – ***Terminaison***

Le processus élu est l'initiateur d'identité minimale, id_*. Une fois élu, il informe tous les processus de son identité et de la fin de l'élection par l'envoi d'un message circulant $\langle \mathbf{fin}, id_* \rangle$ qui termine l'algorithme $\mathcal{A}$.

Institut *Galilée*
année 2004-2005

Ingénierie logicielle ILOG 2
Masters MI1 & MMI1

Algorithmique parallèle et distribuée

Devoir à rendre pour la semaine du 24 janvier 2005

1 Algorithme *CRCW* de calcul d'un maximum

On considère une liste $L = \{s_1, s_2, \ldots, s_n\}$ de n nombres non nécessairement distincts.

1.1 Expliquer en détail comment l'algorithme *Maxrapide*(L) du cours permet de déterminer le maximum de L en temps $O(1)$ sur une PRAM *CRCW commune* de surface $\mathcal{H}(n) = \Omega(n^2)$.

1.2 On souhaite maintenant trouver un algorithme MAX_1 calculant le maximum de L sur une PRAM *CRCW* de surface $\mathcal{H}(n) = n$.

- **a.** Soit une PRAM *CRCW* à p processeurs. Montrer que le problème consistant à trouver le maximum parmi $m \leq p/2$ nombres peut se ramener au problème du calcul du maximum parmi *au plus* m^2/p nombres en $O(1)$ sur une PRAM *CRCW* à p processeurs.
- **b.** Si on commence avec $m = \lfloor p/2 \rfloor$ nombres, combien en reste-t-il après k itérations de Max_1 ?
- **c.** Montrer que MAX_1 peut déterminer le maximum de L en temps $\lceil \lg \lg(n) \rceil$ sur une PRAM *CRCW* à $p = n$ processeurs.
- **d.** Quelle type de PRAM *CRCW* doit utiliser MAX_1 ? Peut-on obtenir les mêmes performances avec un autre type de PRAM ?
- **e.** L'algorithme MAX_1 est-il efficace ? optimal ?

1.3 On suppose maintenant que les n nombres s_i (toujours non nécessairement distincts) de la liste L sont compris entre 1 et n.

- **a.** Donner un algorithme MAX_2 de complexité en temps $T(n) = O(1)$ qui détermine le maximum de L sur une PRAM *CRCW* de surface $\mathcal{H}(n) = n$.
- **b.** Expliquer en quoi la restriction de la taille des nombres permet de trouver un maximum de L en temps constant.
- **c.** Quelle PRAM *CRCW* doit-on utiliser pour MAX_2 ? Peut-on obtenir les mêmes performances avec un autre type de PRAM ?
- **d.** L'algorithme MAX_2 est-il efficace ? optimal ?

2 Algorithme distribué d'élection sur un anneau

Dans la suite, un *système distribué* $\mathcal{S}$ est modélisé par un graphe $G = (\mathcal{P}, \mathcal{L})$ tel que $|\mathcal{P}| = n$ et $|\mathcal{L}| = m$ (cf. cours).

Les hypothèses sont identique à celles de l'algorithme d'élection de Chang-Roberts vu en cours.

- L'algorithmes distribué d'élection $\mathcal{A}$ permet d'élire un processus par *comparaisons* et choix d'un *extremum* sur un anneau de taille n, unidirectionnel, asynchrone et avec identités.
- Chaque processus $P_i \in \mathcal{P}$ $(i = 1, 2\ldots, n)$ possède une identité *unique* id_i qu'il est seul à connaître. Toute identité appartient à l'ordre total $[n] = \{1, \ldots, n\} \subset \mathbb{N}$.
- L'anneau est *asynchrone à délais bornés* (ADB).
- L'algorithmes $\mathcal{A}$ peut être enclenché par tout sous-ensemble non vide $\mathcal{I} \subseteq \mathcal{P}$ des processus *initiateurs*.

L'algorithme $\mathcal{A}$ choisit l'élu à partir de la variable locale *val* définie comme suit :

1. Un processus initiateur d'identité id engendre uniformément et indépendemment un *nombre premier* $id \leq p \leq 2n$, grâce à la fonction aléatoire entière $\mathbf{prem}(id, 2n)$.
2. La variable locale val_i d'un processus P_i $(1 \leq i \leq |\mathcal{I}|)$ est alors calculée comme le produit $id_i \times p$.

2.1 La définition de la variable *val* est-elle mathématiquement fondée ? En d'autres termes (le démontrer),

i) Toutes les variables *val* des processus initiateurs construites à partir de p et id ont-elles bien des valeurs *distinctes* ?

ii) En chaque initiateur d'identité id, la fonction aléatoire entière $\mathbf{prem}(id, 2n)$ a-t-elle un bien sens ? Autrement dit, engendre-t-elle un *nombre premier* p tel que $id \leq p \leq 2n$ grâce à un tirage aléatoire *uniforme* sur un ensemble d'entiers défini correctement et *sans ambiguïté* ?

2.2 Expliquer en détail le déroulement de l'algorithme distribué $\mathcal{A}$ spécifié ci-dessus en prenant $1 < |\mathcal{I}| = \ell < n$. Préciser le comportement des ℓ initiateurs et celui des $n - \ell$ processus non initiateurs, ainsi que le rôle joué par la variable *val*.

Montrer que l'algorithme $\mathcal{A}$ élit un processus qui n'est *pas nécessairement* celui qui serait élu par l'algorithme de Chang-Roberts.

2.3 Complexité en messages. Quelle sont les configurations qui constituent le *meilleur cas* et le *pire cas* pour la complexité en messages de l'algorithme $\mathcal{A}$? En déduire ces deux mesures de complexité : $C_{min}(n)$ et $C_{max}(n)$,

1^o pour un initiateur *unique*,

2^o lorsque *tous les processus* sont initiateurs.

En déduire la complexité *moyenne* en messages $\overline{C(n)}$. de l'algorithme $\mathcal{A}$

2.4 Complexité en temps. Quelle est la configuration qui constituent le *pire cas* pour la complexité en temps $T(n)$ de $\mathcal{A}$? Calculer $T(n)$.

2.5 Complexité en espace-mémoire. Calculer la complexité en mémoire $S(n)$ de $\mathcal{A}$ dans le pire des cas, lorsque *tous les processus* sont initiateurs. Quelle est la valeur de cette complexité lorsque $1 \leq |\mathcal{I}| = \ell < n$.

2.6 Comparer les mesures de complexité respectives (en messages, en temps et en espace-mémoire) de l'algorithme d'élection $\mathcal{A}$ et de celui de Chang-Roberts. Qu'en concluez-vous ?

Algorithme $\mathcal{A}$ (pour un processus P_i quelconque)

ÉTAPE 1 – ***Élection***

var $état_i$: {*élu*,*battu*,*candidat*,*passif*} (initialisée à *passif*) ;
val : entier ;
$Liste_i$: sous-ensemble de $\mathbb{N}$ (initialisée à $\{val_i\}$) ;

début si P_i est initiateur **alors**
$p := \mathbf{prem}(id, 2n)$; $val_i := id_i \times p$; $état_i := candidat$;
envoyer $\langle val_i \rangle$; recevoir $\langle val \rangle$;
tantque $val_i \neq val$ **faire**
$Liste_i := Liste_i \cup \{val\}$; envoyer $\langle val \rangle$; recevoir $\langle val \rangle$;
fintantque ;
si $val_i = \max(Liste_i)$ **alors** $état_i := élu$
sinon $état_i := battu$ **fsi**
sinon **répéter** recevoir $\langle val \rangle$; envoyer $\langle val \rangle$;
si $état_i = passif$ **alors** $état_i := battu$ **fsi**
jusqu'à faux
fsi ;
fin

ÉTAPE 2 – ***Inauguration-Terminaison***

*Le processus initiateur élu est celui de valeur maximale, val_{max}, lisible dans sa liste Liste. Une fois élu, il informe tous les processus de son identité et de la fin de l'élection par l'envoi d'un message circulant $\langle$**fin**,$val_{max}\rangle$ qui termine l'algorithme $\mathcal{A}$.*